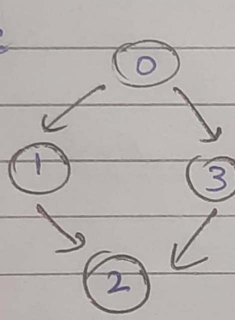
Topological Sort

DAG ⇒ Directed Acyclic Graph } Directed होगा और cycle exist नहीं करेगी ॥

** • Topological Sort is applicable on DAG only

* topological sort of DAG is the linear Ordering of vertices such that for every directed edge from vertex "u" to vertex "v", vertex "u" comes before "v" in the ordering

ex:-



for path/ordering:

[0 1 3 2] So this Valid Top. Sort?

⇒ In this ordering every "u" to "v" edge.

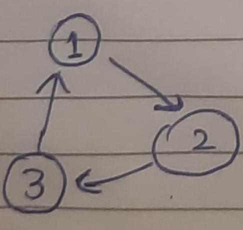
⇒ "u" comes before "v"

☆ Yes its Valid top. Sort

adj list

0 → 1,3
1 → 2
3 → 2
2 → _

Q Why top. Sort only in DAG?



adj list
1→2
2→3
3→1

⇒ ordering :- [1 | 2 | 3]

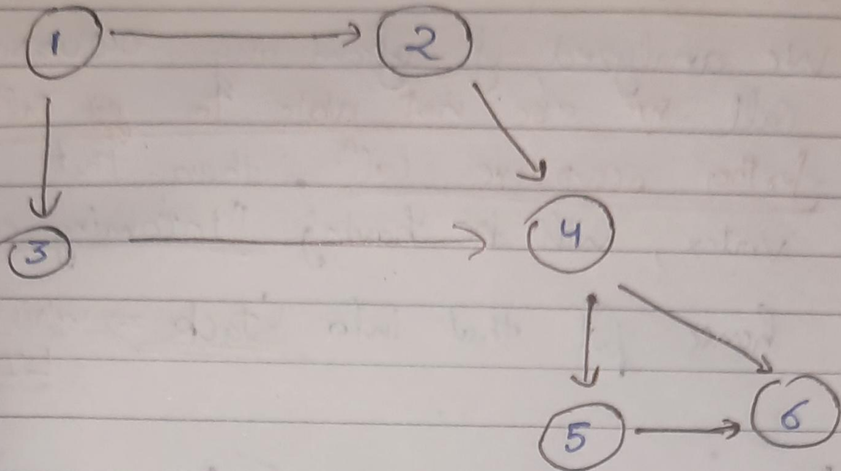
here; (3,1) will result in Invalid Top sort.

So, will exist at least one edge in Cyclic Graph which will give invalid ORDERING

Top. sort (by DFS)

Date : / /

Page No.



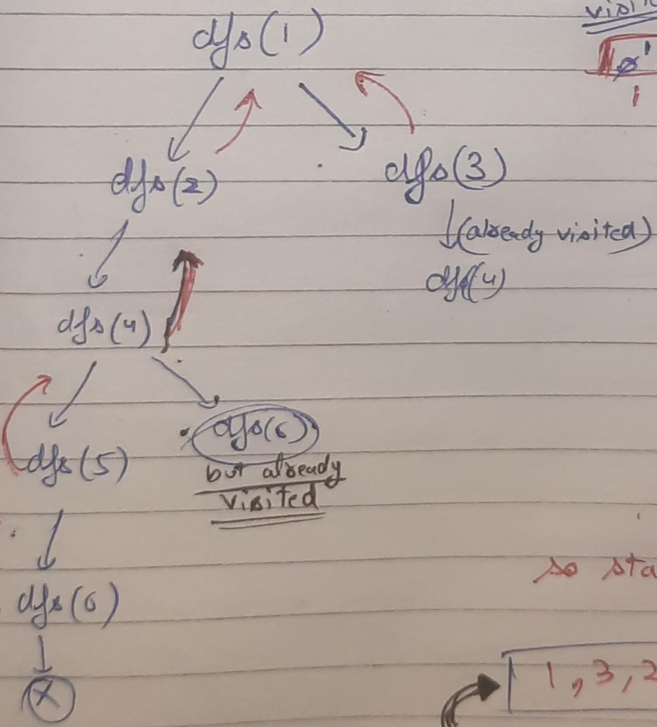
for DFS { • So, we will need a "visited" data structure

for top'. sort { • We need additional D.S ⇒ Stack

adj list

1→	2,3
2→	4
3→	4
4→	5,6
5→	6
6 →	—

dry run



1
3
2
4
5
6

Stack

visited

1	2	3	4	5	6
~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1	~~0~~ 1

Note: whenever we backtrack from a call then, push into stack

so stack order

⇓

1, 3, 2, 4, 5, 6

Valid topological sort

Date : / /

Why need of Stack in Topological Sort ?

⇒ We analyzed, if from any recursive call we are not able to call further recursive call, than that vertex, will be having "incoming edge" hence put that into Stack ⇒ which shows LIFO